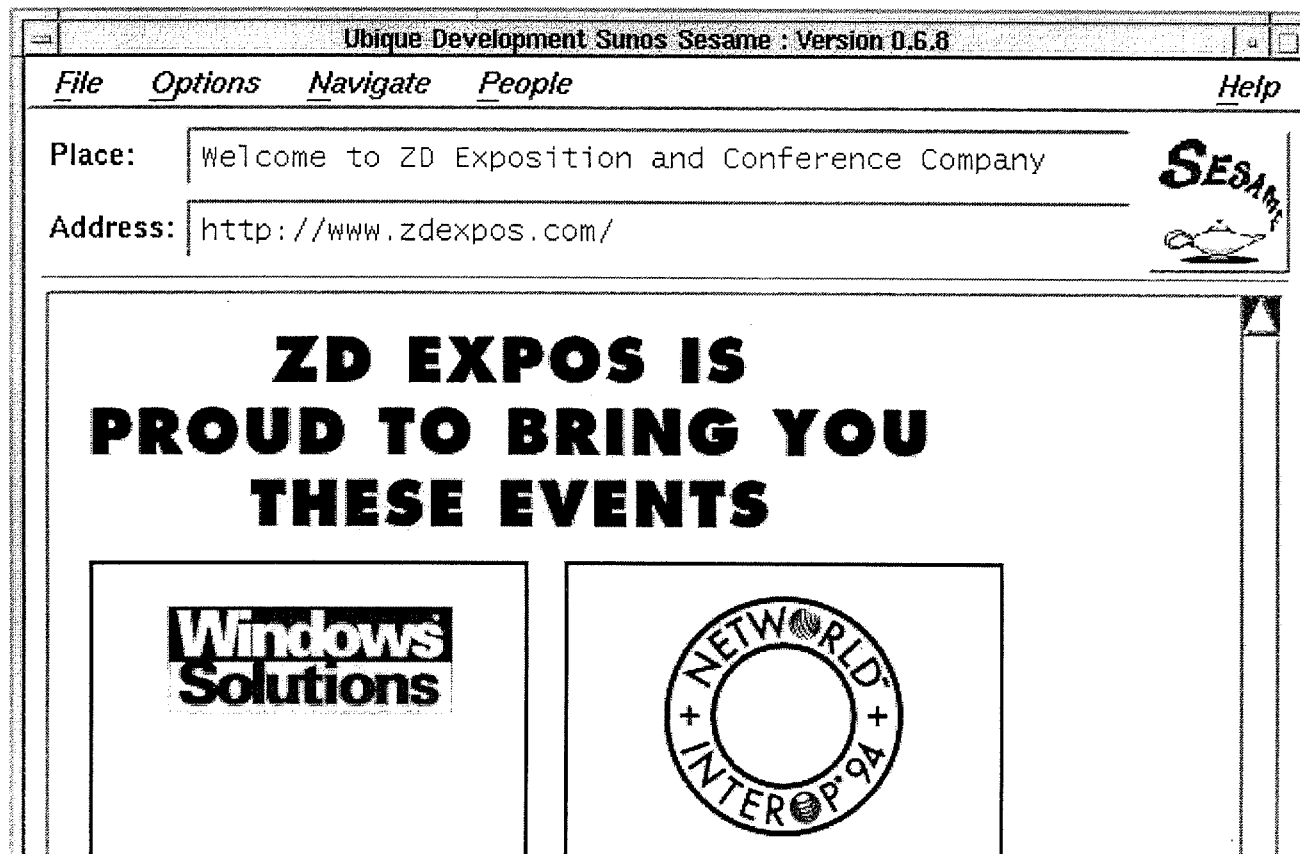


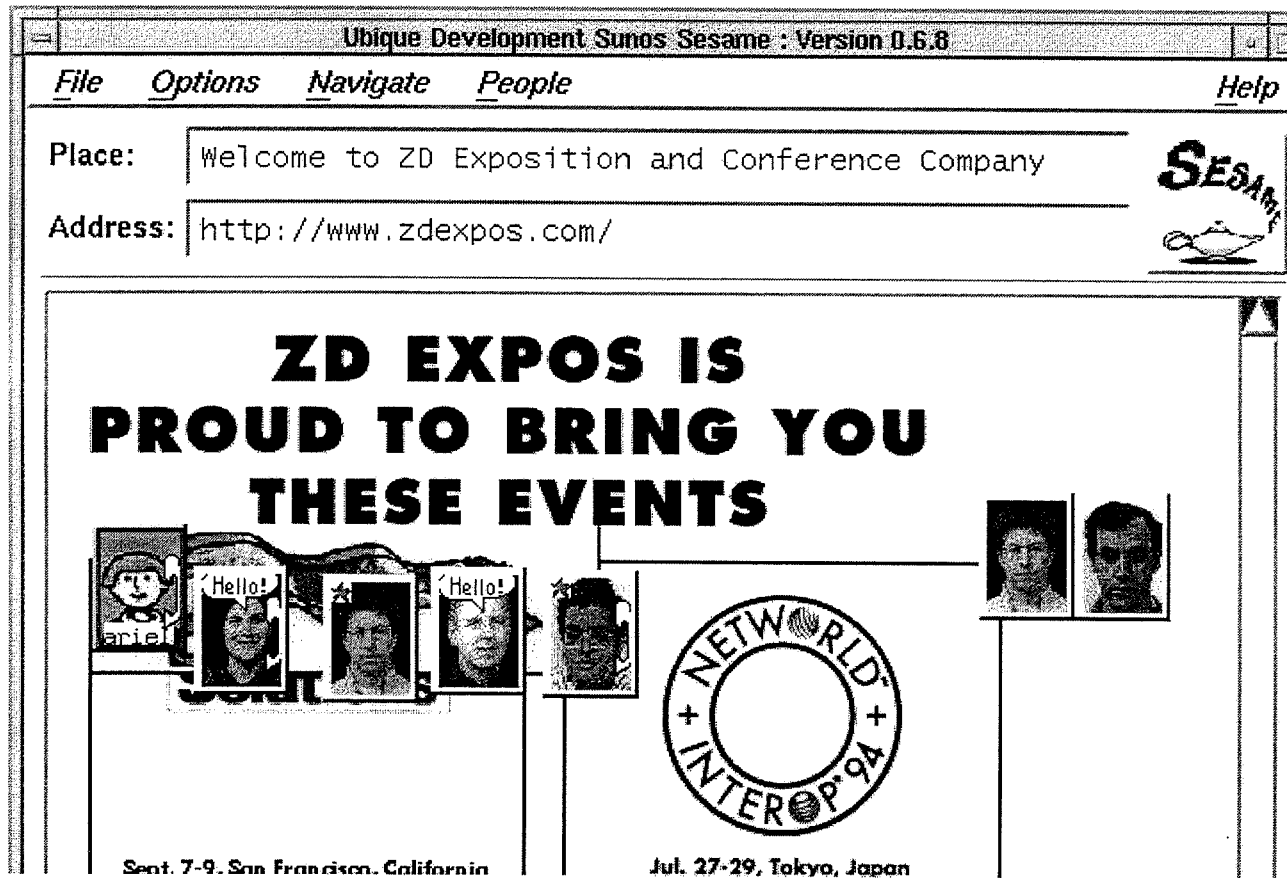
# ***The World Wide Web: a new communication medium***

- ☐ **Thousands of servers and growing**
- ☐ **Millions of users**
- ☐ **Multiple applications**
  - Marketing venue
  - On-line publication
  - Point of sale
  - Customer service
  - Community Center

*Hello, anybody home ?*

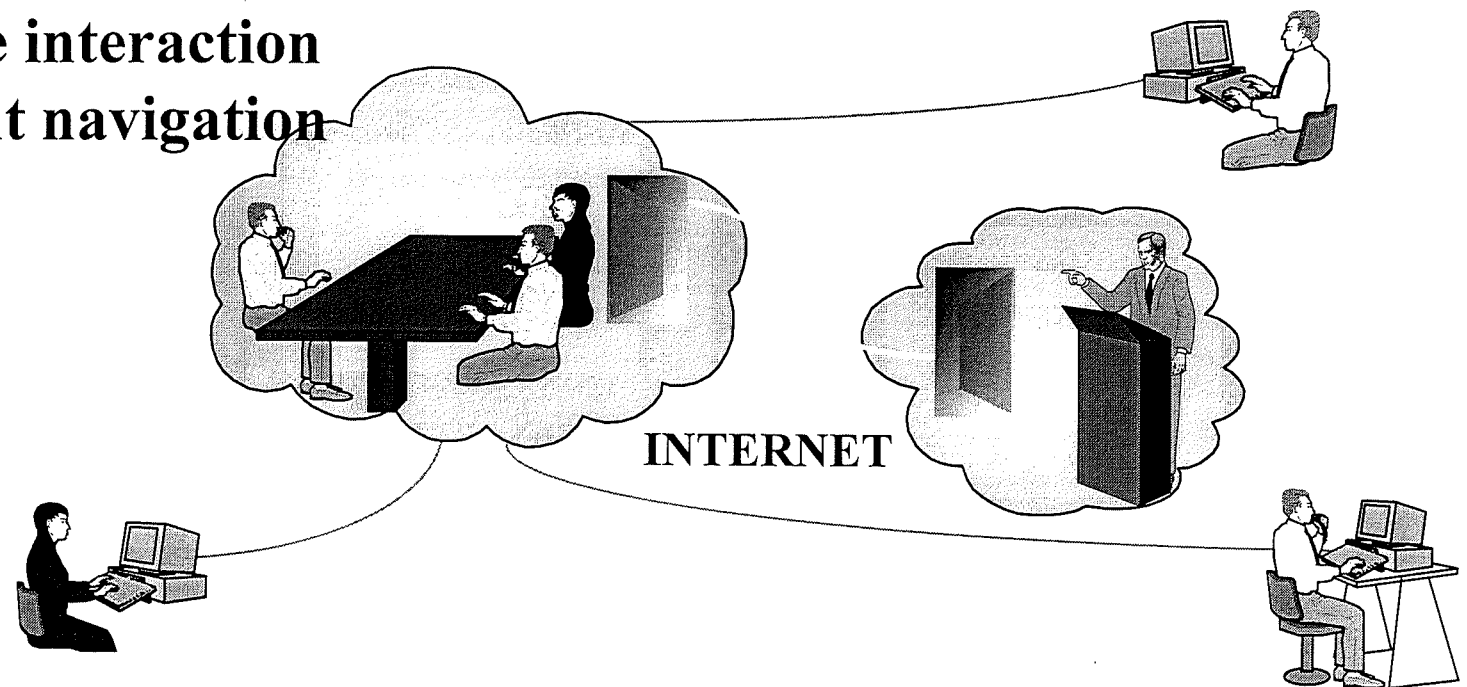


# *Virtual Places<sup>TM</sup> -- adding people to the Web*



# *Virtual Places: Web Info + Human Presence*

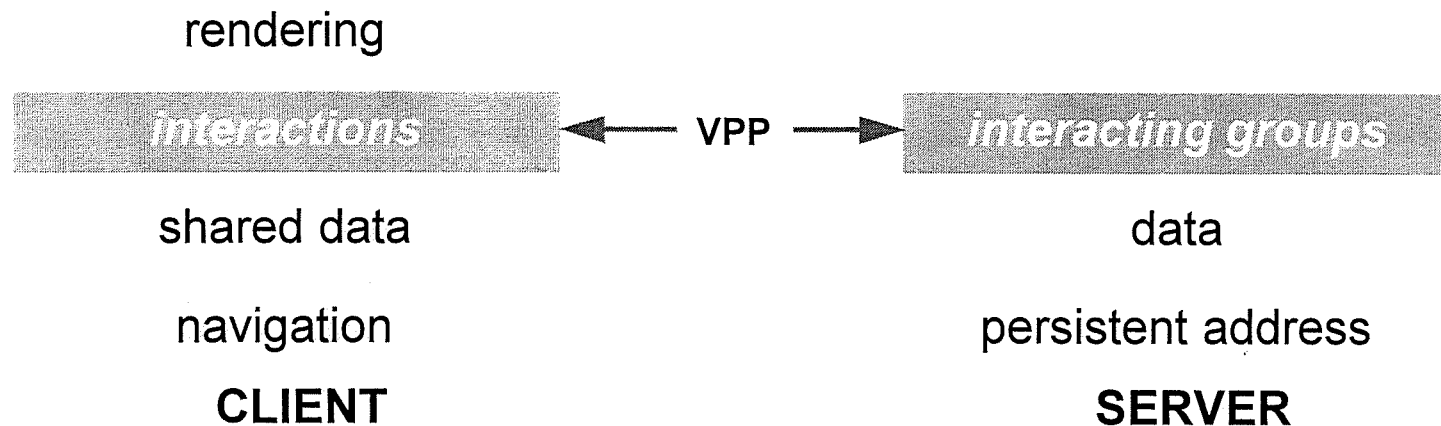
- ☐ Web Information, combined with
- ☐ Human presence
- ☐ Live interaction
- ☐ Joint navigation



# *Virtual Places architecture\**

- ❑ **Modular and open client and server extension to the Web**
- ❑ **Supports many forms of live human interaction**
  - one-to-one, groups, lectures, guided tours
- ❑ **Intuitive point-and-click interface**
  - join/leave, initiate/terminate, navigate

*\*patent pending*



# ***Applications for AOL***

- ☐ **“Personal virtual home” for every member**
- ☐ **Guided tours of AOL space/the Web**
- ☐ **Sponsored on-line events**
  - At AOL content
  - At any Web server on the Internet
- ☐ **Staffed shopping malls**

## ***Personal home: a web page turned virtual place***

- ☐ **Put your own content**
- ☐ **Host your friends**
- ☐ **“Go out” with friends (starting point for private tours)**

# ***AOL-Initiated Public Events***

- ☐ **Enrich the experience using on-line events**
- ☐ **Theme-based guided tours of the entire Web, e.g.**
  - Museums and Galleries
  - Corporations
  - Universities
  - Wall Street
  - Government
- ☐ **Lectures, conferences, debates**
  - Incorporating Web content



# *Product Status*

*Avner Shafrir*  
*VP of R&D, Ubique*

# *Main R&D goals during 95*

## **☐ March 95 - Windows Alpha / UNIX final release**

- Windows 3.1 / 32 bit
- Single VP server
- Basic functionality

## **☐ End of November 95 - Final Release**

- All Windows platforms, Mac
- Multi-servers
- Extensions

## ***Major events during 95***

**We have modified the Virtual Places model along 95:**

**☐ March 95 - N+I demo & Alpha release**

- 10,000 Virtual Places installations
- Usability study

**☐ The on-going process with AOL**

- Integration with AOL-client (May 95)
- Community/on-line services model

**But the main schedule was not changed**

# *Virtual Places*

**Virtual Places is a cross platform application environment supporting casual real-time communication and collaboration among people in communities over the World Wide Web.**

# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
  - Windows
  - Macintosh
- ☐ **application environment**
- ☐ **supporting casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
- ☐ **application environment**
  - Toolbox
  - Architecture
  - Extension to the Web
  - Desk top level (MSoffice)
- ☐ **supporting casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ❑ Virtual Places is a **cross platform**
- ❑ **application environment**
- ❑ **supporting casual**
  - everyday
  - non-business oriented
- ❑ **real-time**
- ❑ **communication**
- ❑ **and collaboration**
- ❑ **among people**
- ❑ **in communities**
- ❑ **over the World Wide Web.**

# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
- ☐ **application environment**
- ☐ **supporting casual**
- ☐ **real-time**
  - co-presence
  - capacity
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**



# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
- ☐ **application environment**
- ☐ **supporting casual**
- ☐ **real-time**
- ☐ **communication**
  - text chat
  - audio
  - video
  - broadcasting
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
- ☐ **application environment**
- ☐ **supporting casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
  - guided tour
  - document sharing
  - extensions; applets
  - games
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ☐ Virtual Places is a **cross platform**
- ☐ **application environment**
- ☐ supporting **casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
  - one-to-one, 1-many, multiway
  - pictures, animation, 3D avatars
  - human extensions - (intelligent agent)
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ☐ Virtual Places is a **cross platform**
- ☐ **application environment**
- ☐ supporting **casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
  - privately ; publicly in groups
  - large capacity
  - community services
- ☐ **over the World Wide Web.**

# *Virtual Places*

- ☐ Virtual Places is a **cross platform**
- ☐ **application environment**
- ☐ supporting **casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**
  - In context
  - At scheduled events

# *Virtual Places*

- ☐ **Virtual Places is a cross platform**
- ☐ **application environment**
- ☐ **supporting casual**
- ☐ **real-time**
- ☐ **communication**
- ☐ **and collaboration**
- ☐ **among people**
- ☐ **in communities**
- ☐ **over the World Wide Web.**

# *Vision & Benefits*

## **□ End Users**

- As an end-user application, it lets people meet, interact and communicate easily and directly within the context provided by the Web.

## **□ Service Provider**

- As a communication channel, it enhances the human dimension of the rich content being published over the Internet, providing a way to bind people together (customer-vendor, author-publisher, friends and colleagues).

## **□ Application Environment**

- As open architecture/platform, it encourages creativity by means of open extensibility, customization, and application development.

# *Vision & Benefits*

## **❑ End Users**

- As an end-user application, it lets people meet, interact and communicate easily and directly within the context provided by the Web.

## **❑ Service Provider**

- As a communication channel, it enhances the human dimension of the rich content being published over the Internet, providing a way to bind people together (customer-vendor, author-publisher, friends and colleagues).

## **❑ Application Environment**

- As open architecture/platform, it encourages creativity by means of open extensibility, customization, and application development.



# *Vision & Benefits*

## **❑ End Users**

- As an end-user application, it lets people meet, interact and communicate easily and directly within the context provided by the Web.

## **❑ Service Provider**

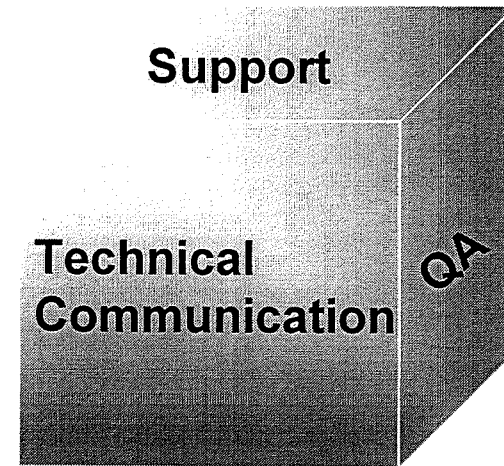
- As a communication channel, it enhances the human dimension of the rich content being published over the Internet, providing a way to bind people together (customer-vendor, author-publisher, friends and colleagues).

## **❑ Application Environment**

- As open architecture/platform, it encourages creativity by means of open extensibility, customization, and application development.

## *Ubique R&D - Structure & Tasks*

- ❑ Virtual Places - Architecture
- ❑ Virtual Places - Client
- ❑ Virtual Places - Technologies



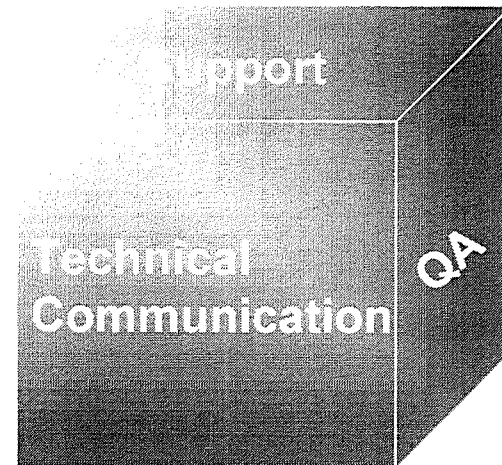
# *Ubique R&D - Structure & Tasks*

## ☐ Virtual Places - Architecture

- Protocol & Communication
- Multi Servers
- Functional Extensibility
- Shared Code Base
- API

## ☐ Virtual Places - Client

## ☐ Virtual Places - Technologies



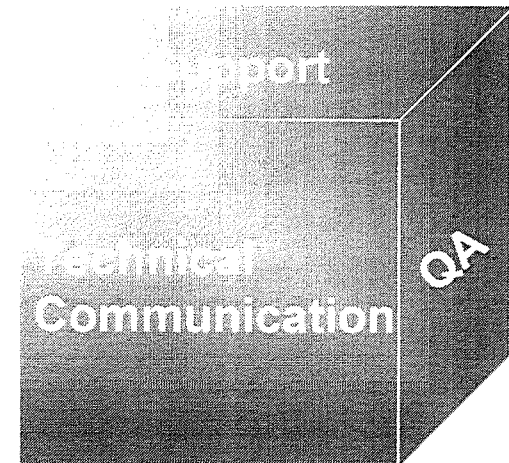
# *Ubique R&D - Structure & Tasks*

## ☐ Virtual Places - Architecture

## ☐ Virtual Places - Client

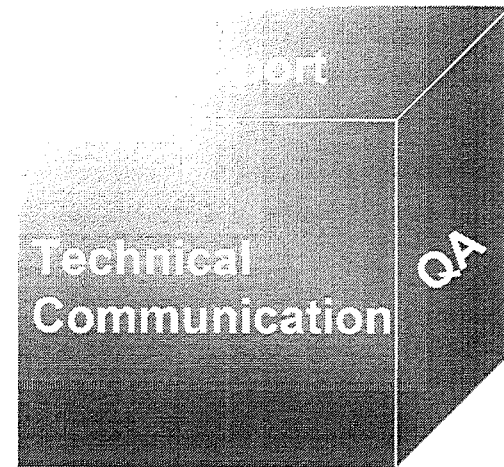
- Client Architecture
- Features
- UI
- Porting

## ☐ Virtual Places - Technologies



# *Ubique R&D - Structure & Tasks*

- ❑ Virtual Places - Architecture
- ❑ Virtual Places - Client
- ❑ Virtual Places - Technologies
  - Audio
  - Video
  - Animation
  - VRML
  - Intelligent Agents / Extensions



# *Ubique R&D - Structure & Tasks*

## ☐ Virtual Places - Architecture

- Protocol & Communication
- Multi Servers
- Functional Extensibility
- Shared Code Base
- API

## ☐ Virtual Places - Client

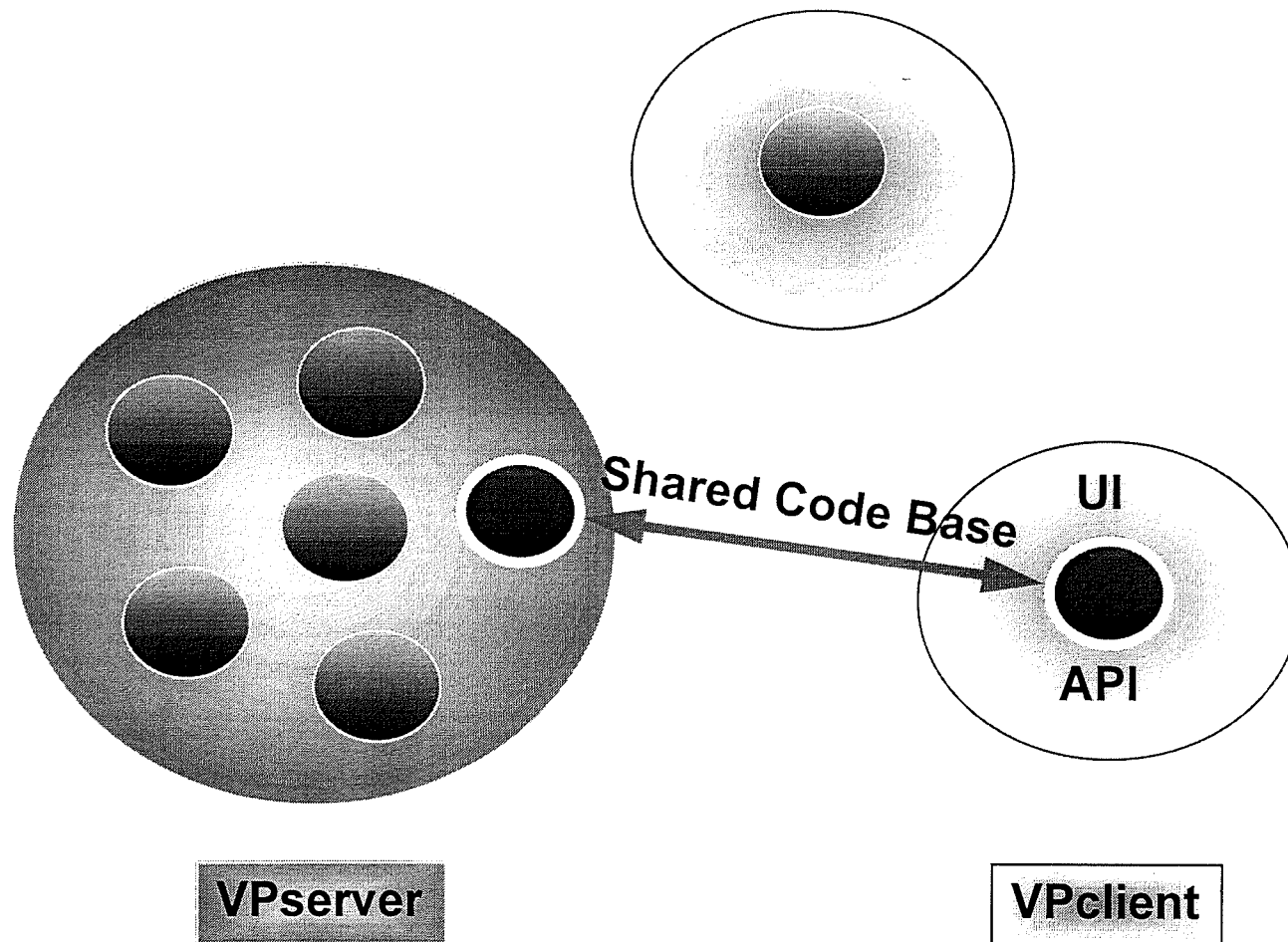
## ☐ Virtual Places - Technologies

**Support**

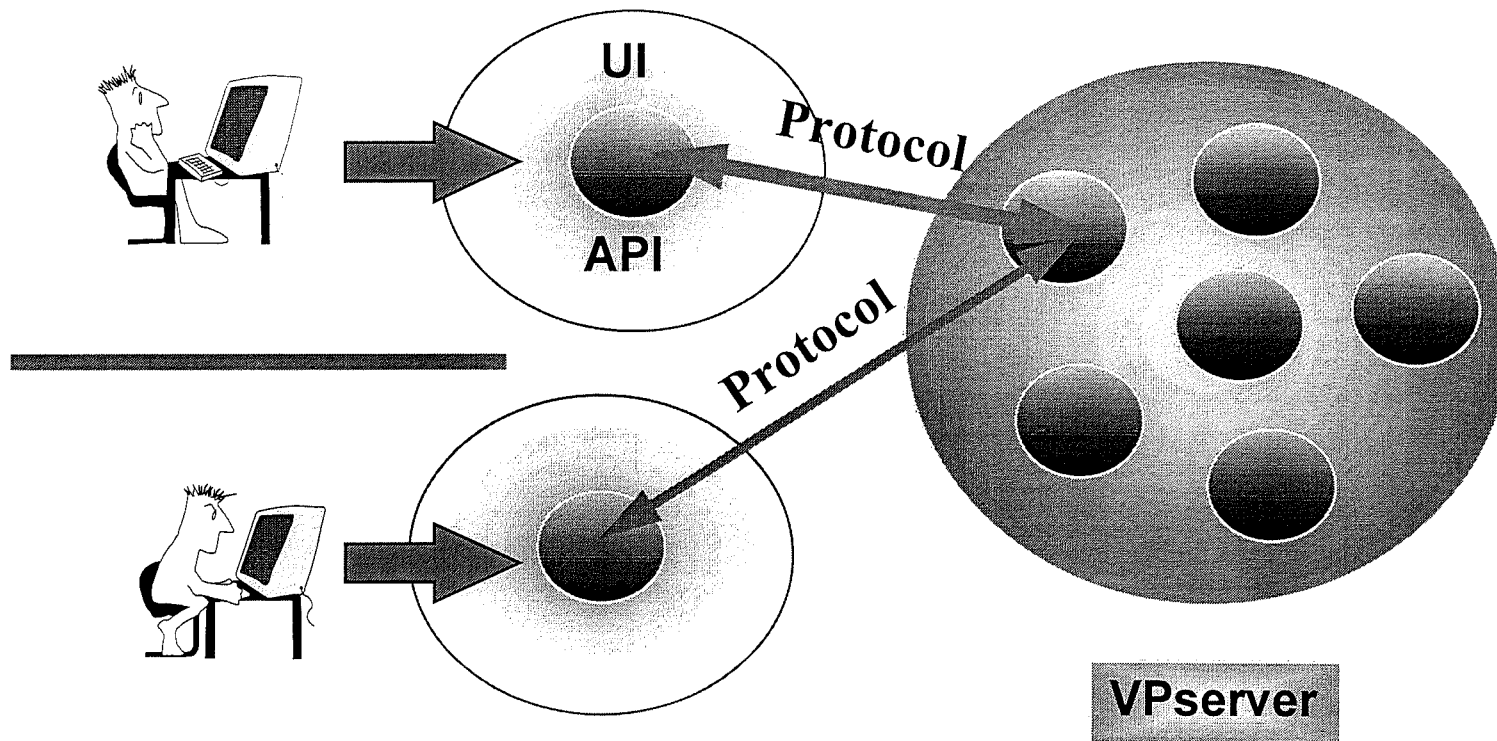
**Technical  
Communication**

**QA**

# *Architecture - Shared Code Base*



# *Architecture - Shared Code Base*





# *Architecture - API to the UI*

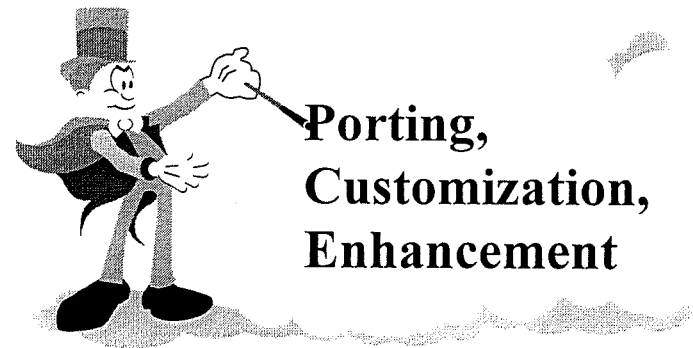
## ☐ Easy porting to new platforms

- Windows 95
- Mac
- Unix (?)

## ☐ Customized UI

## ☐ Enhancement

- Animation
- Games
- Intelligent Agents



# *Architecture - Server*

## ☐ **Groups Server**

- One on One group
- Tour
- Lecture
- Ad-hoc groups / Application sharing

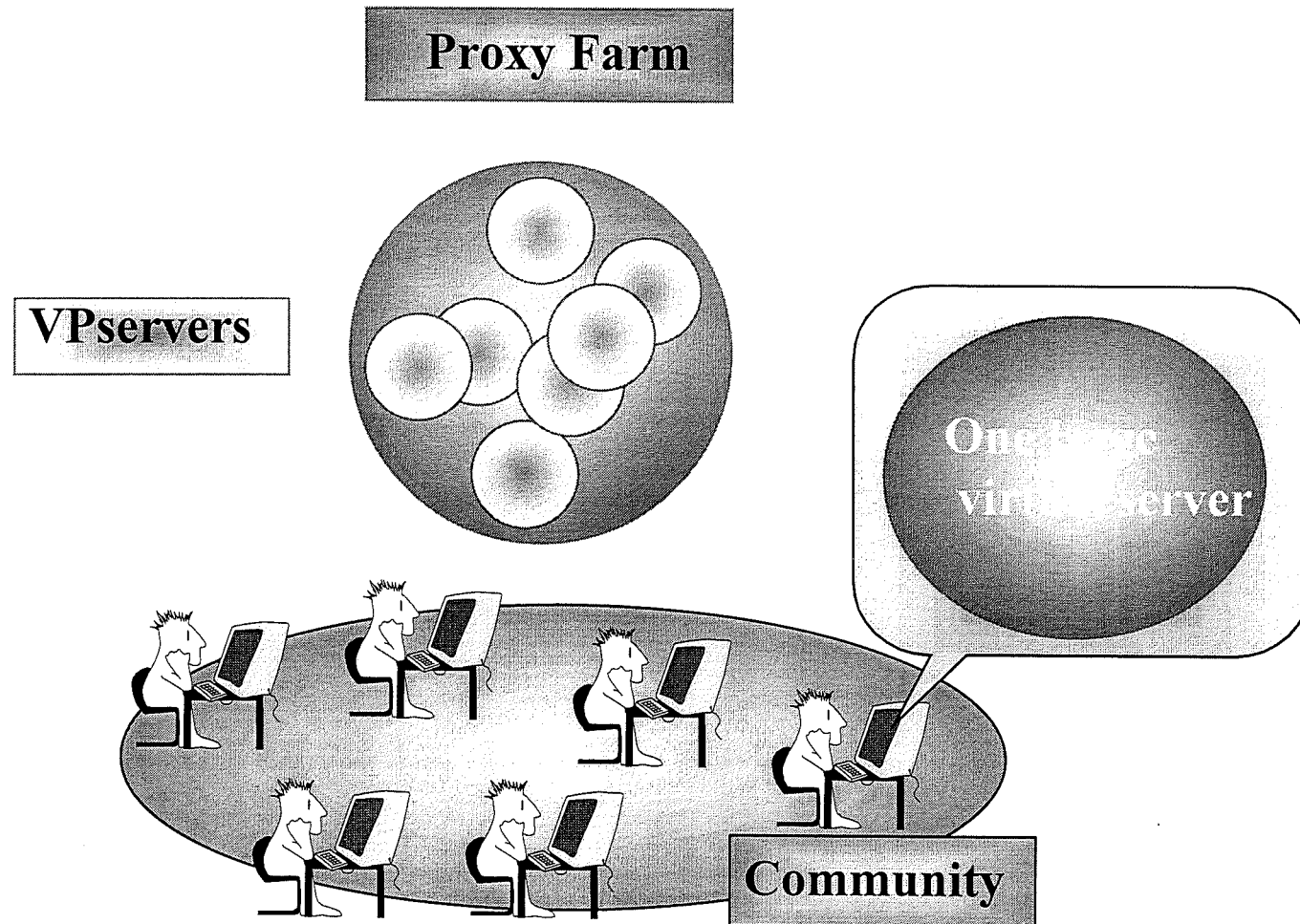
## ☐ **“Data Base” Server (Attributes)**

- Group name
- Objects

## ☐ **Multi server**

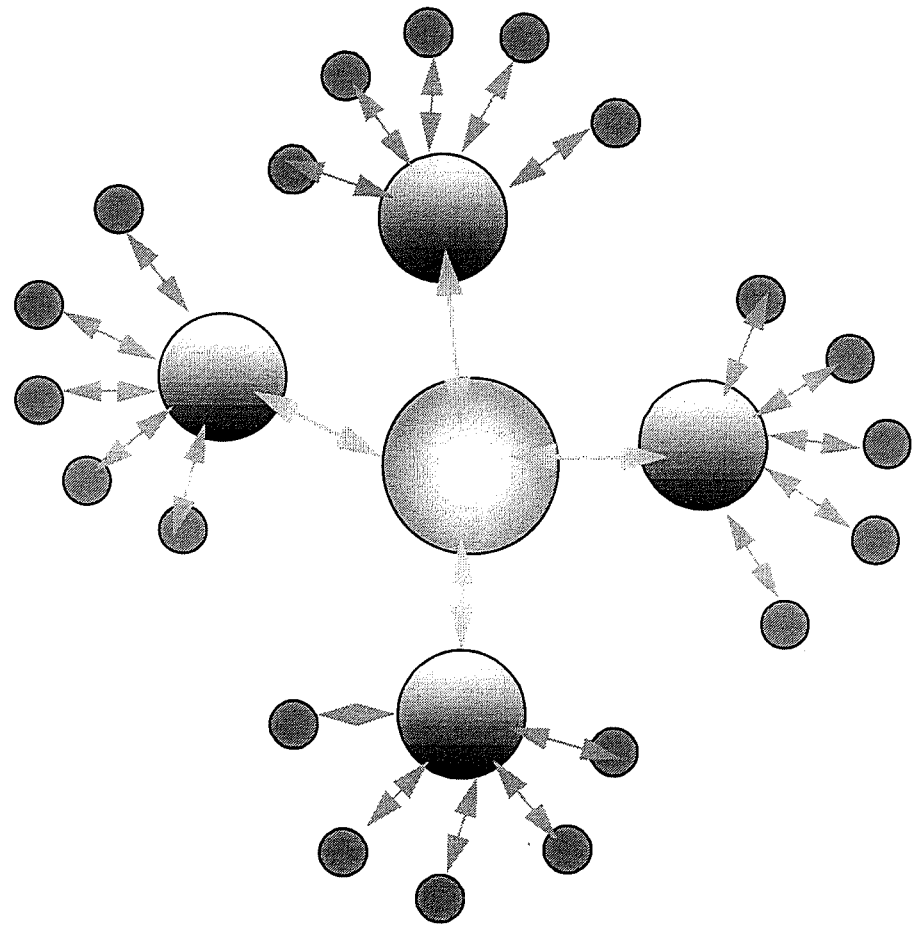
- Shared access to any Web server
- Load Balancing

# *Architecture - Multi Server*



# *Architecture - Multi Server Model*

- ❑ Load of presences should be shared between many servers to create huge communities
- ❑ Management problems
  - Finding/Creating places
  - Finding a presence
  - Broadcasting

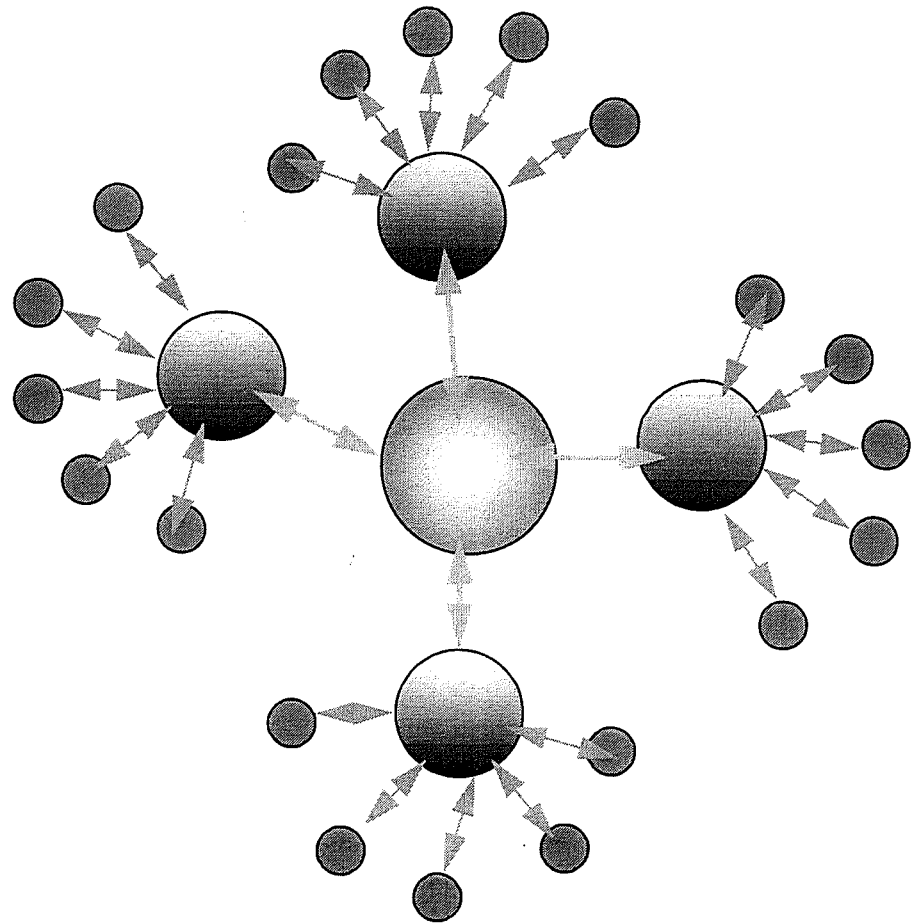


# *Architecture - Multi Server Model*

Virtual Places Servers

Cluster Manager

Cell Manager



# *Server Types*

**Virtual Places Servers**

**Cluster Manager**

**Manager**

## **Virtual Places Functionality**

**Server to Server connection**

**Creation of new places by Manager demand**

**Sends updates to the manager e.g. presence leaves**

**Queries to Manager**

# *Server Tasks*

**Virtual Places Servers**

## **Virtual Places Functionality**

**Server to Server connection**

**Creation of new places by Manager demand**

**Sends updates to the manager e.g presence leaves**

**Queries to Manager**

**Cluster Manager**

**Keeps updated state of all the VP servers**

**Allocation for new places (load balancing)**

**Queries and answers with Central Server**

**Central Server**

# *Server Tasks*

**Virtual Places Servers**

**Virtual Places Functionality**

**Server to Server connection**

**Creation of new places by Manager demand**

**Sends updates to the manager e.g presence leaves**

**Queries to Manager**

**Cluster Manager**

**Keeps updated state of all the VP servers**

**Allocation for new places (load balancing)**

**Queries and answers with Central Server**

**Nominates Cluster Managers to open new places**

**Distribution node of Cluster Manager queries**



# *Capacity Analysis*

- ❑ 100 Virtual Place servers with 3000 presences each  
(depends on the machine capability)
- ❑ 10 Clusters, 300,000 presences per cluster
- ❑ Total of 3,000,000 presences in this community

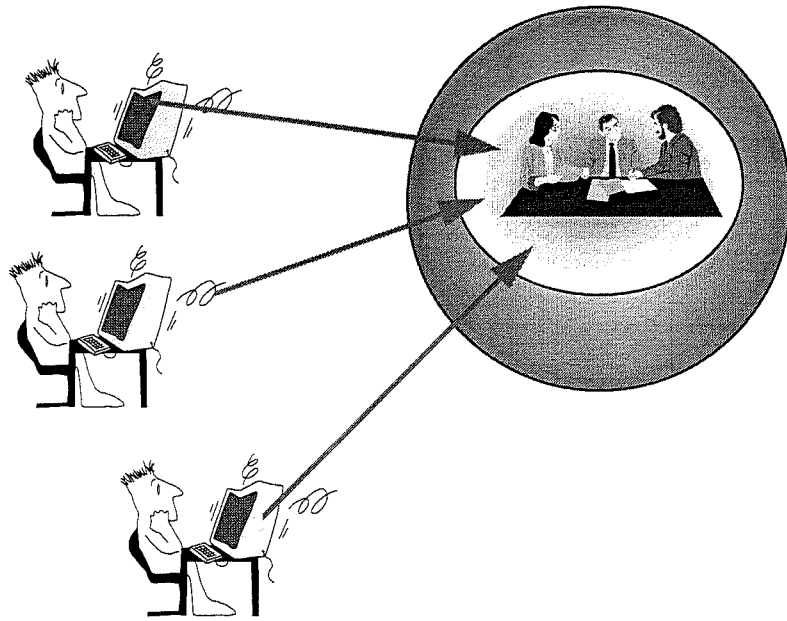
X

**1011 machines to create a Virtual Place Server that supports 3,000,000 people community.**

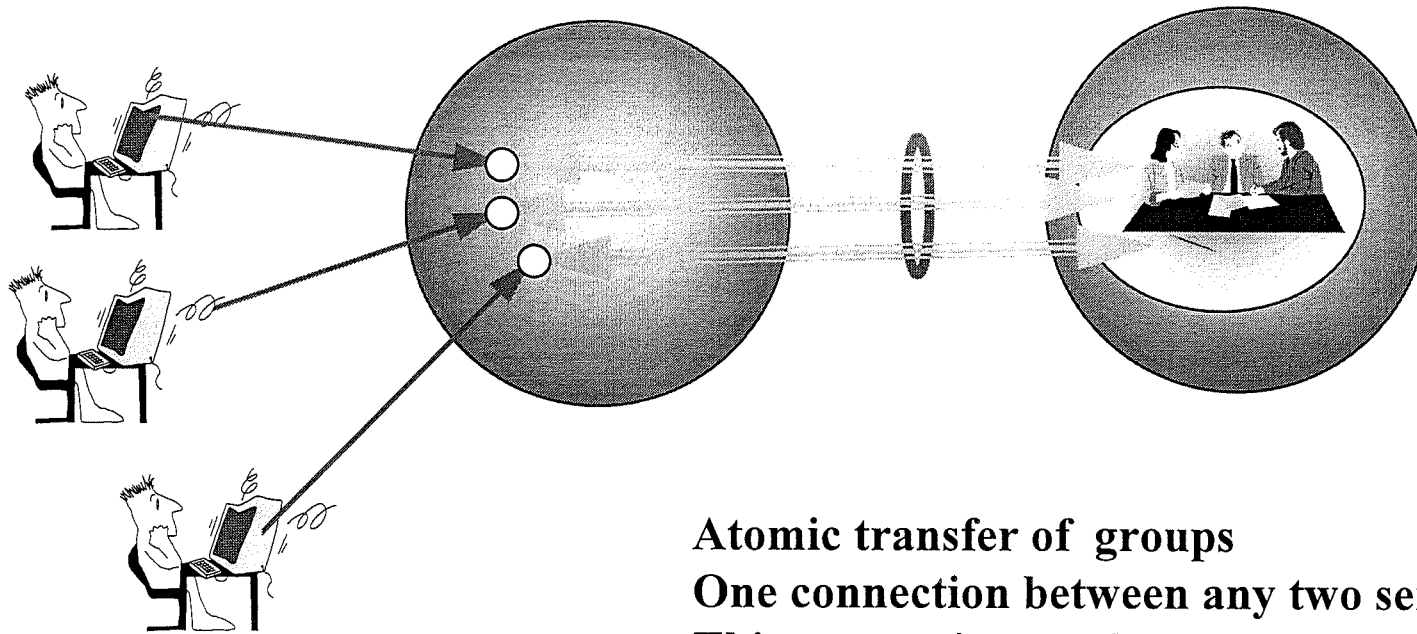
# *Benefits*

- ❑ Community queries can be answered by 1-6 hops (without optimization)
- ❑ Modular/dynamic enhancement of capacity
- ❑ Simple algorithms and decision making — FOR DYNAMIC ALGOS
- ❑ Full control/monitoring and data retrieval about the community via the central server.
- ❑ Simple recovery mechanism - “reactivate and update your state”

# *Server To Server Connection*

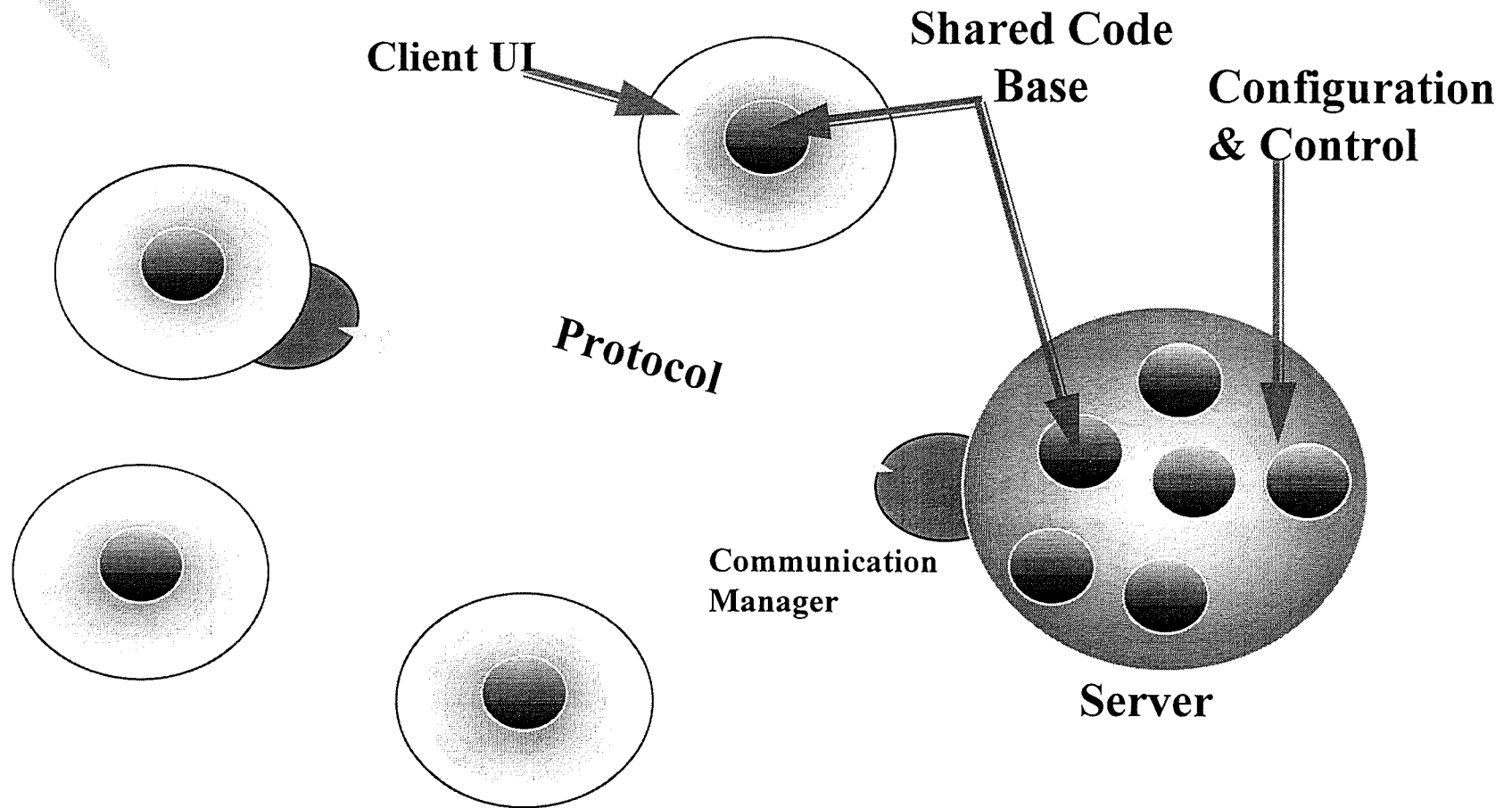


# *Server To Server Connection*



**Atomic transfer of groups**  
**One connection between any two servers**  
**This connection can be permanently open.**

# *Architecture Scheme*



# *Communication Manager*

- ☐ **Manages client/server communication**
  - Can support different communication protocols
- ☐ **A separate module --- easily replaced**
- ☐ **Allows the server to handle many clients**
- ☐ **Optimizes network usage**

## *Virtual Place Architecture*

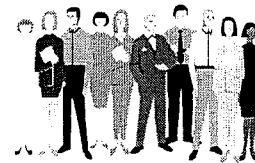
- ☐ Places contain presences and groups
- ☐ Presences are created by clients
- ☐ Places are updated according to presences' requests
- ☐ Clients mirror the places of their presences
- ☐ The server controls the state of a place
- ☐ A client accepts a snapshot when it enters a place
- ☐ Incremental place state changes are propagated to clients with presences in the place

# *Typical Events*

☐ Connecting to a place



☐ Creating/joining a group



☐ Moving between groups





## *Connecting to a Place*



- ❑ Client *C* sends connect request
- ❑ Server adds presence *P* of *C* to the place
- ❑ Server sends snapshot to *C*
- ❑ Server propagates “presence-entered(*P*)” event
- ❑ Client *C* creates (local) place from snapshot
- ❑ Clients add *P* to (local) place

# *Creating/Joining a Group*



❑ Presence  $P$  sends request to server

❑ If request approved by server:

- Server updates place state
- Server propagates an event
- Clients update (local) place state

❑ If request denied by server:

- Reject message is sent to  $P$

## ***Moving between groups***

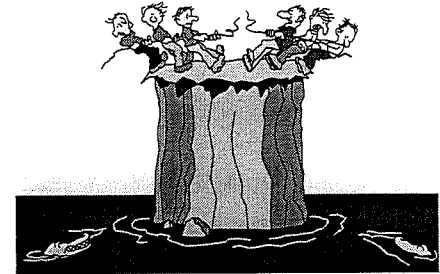


- ☐ Presence  $P$  sends a request to move into a group  
(and position/slot within this group)
- ☐ Server updates place state
- ☐ Server propagates an event
- ☐ Clients update place state

# *Place Capacity*

## ❑ Requirements

- Limited by resources and to preserve clarity
- On the other hand, we want to enable everyone to at least view the place



## ❑ Implementation

- A place is composed of a *room* and a *corridor*.
- Active presences - inside the *room* (limited)
- Passive presences - in the *corridor* (unlimited)
  - Can view place
  - Can enter the room
  - Cannot interact with other presences

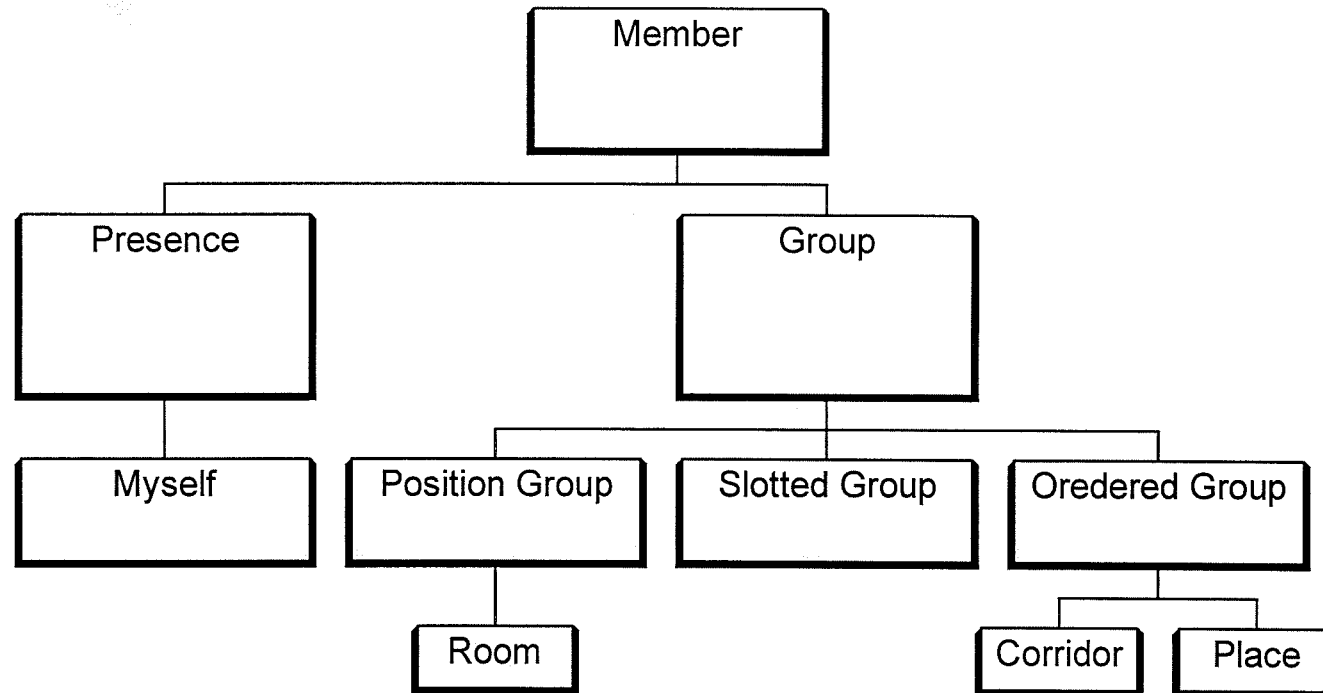
## ❑ Optimization

Presences are not aware of other passive presences

# ***VP Implementation***

- ❑ Object Oriented, in C++**
- ❑ Places are composed of presences and groups**
- ❑ Place, Room & Corridor are special kind of groups**

# *VP Class Hierarchy*



The application can define its derived classes

# ***VP Interface***

- ☐ **Main object is Place**
- ☐ **Client creates a Place instance per window**
  - Along with a “room”, a “corridor” and a “myself”
- ☐ **Interface from application to VP via methods of VP classes**
- ☐ **Interface from VP to application via virtual methods**
  - Informing on state change events
  - Specifically, creating places members (groups and presences)

# *Ubique R&D - Structure & Tasks*

## ☐ Virtual Places - Architecture

## ☐ Virtual Places - Client

- Client Architecture
- Features
- UI
- Porting

## ☐ Virtual Places - Technologies

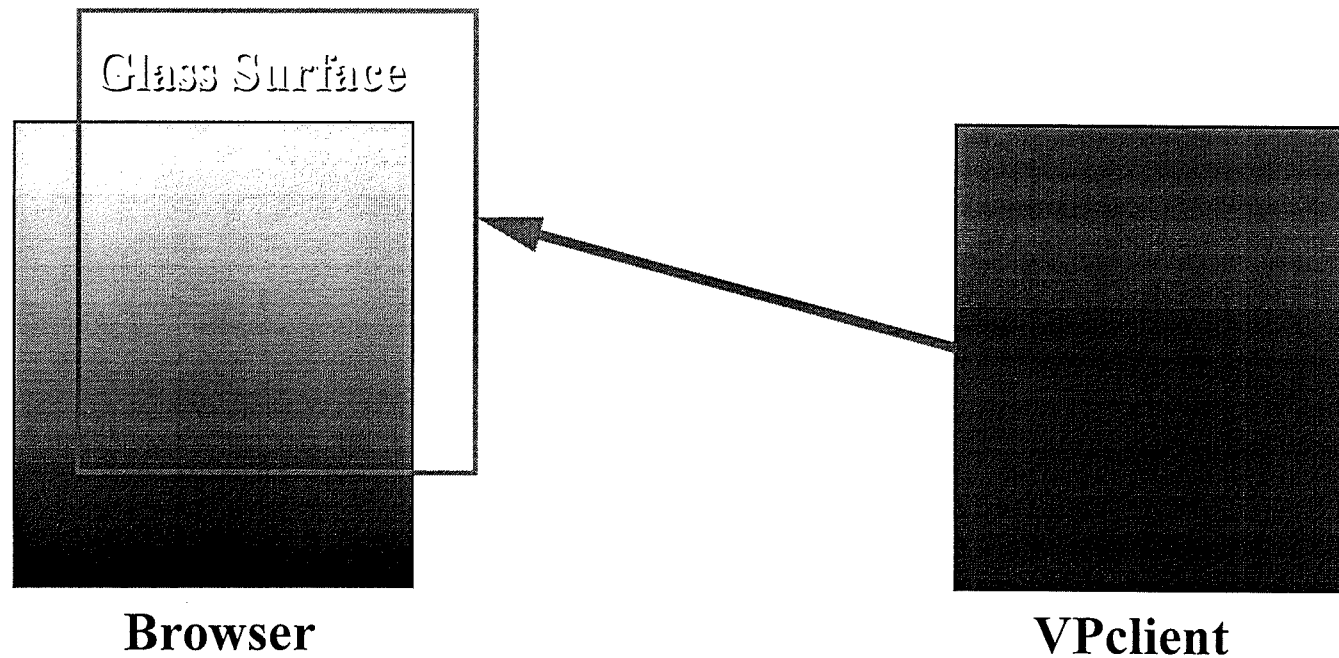
**Support**

**Technical  
Communication**

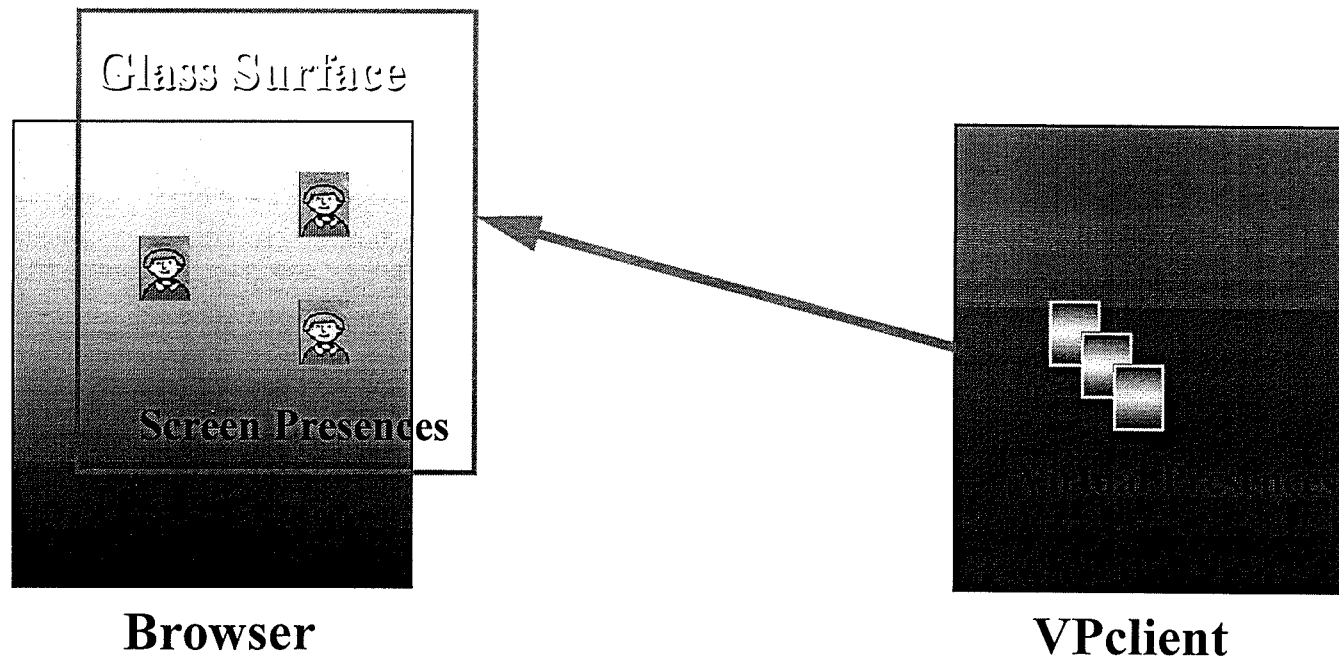
**QA**



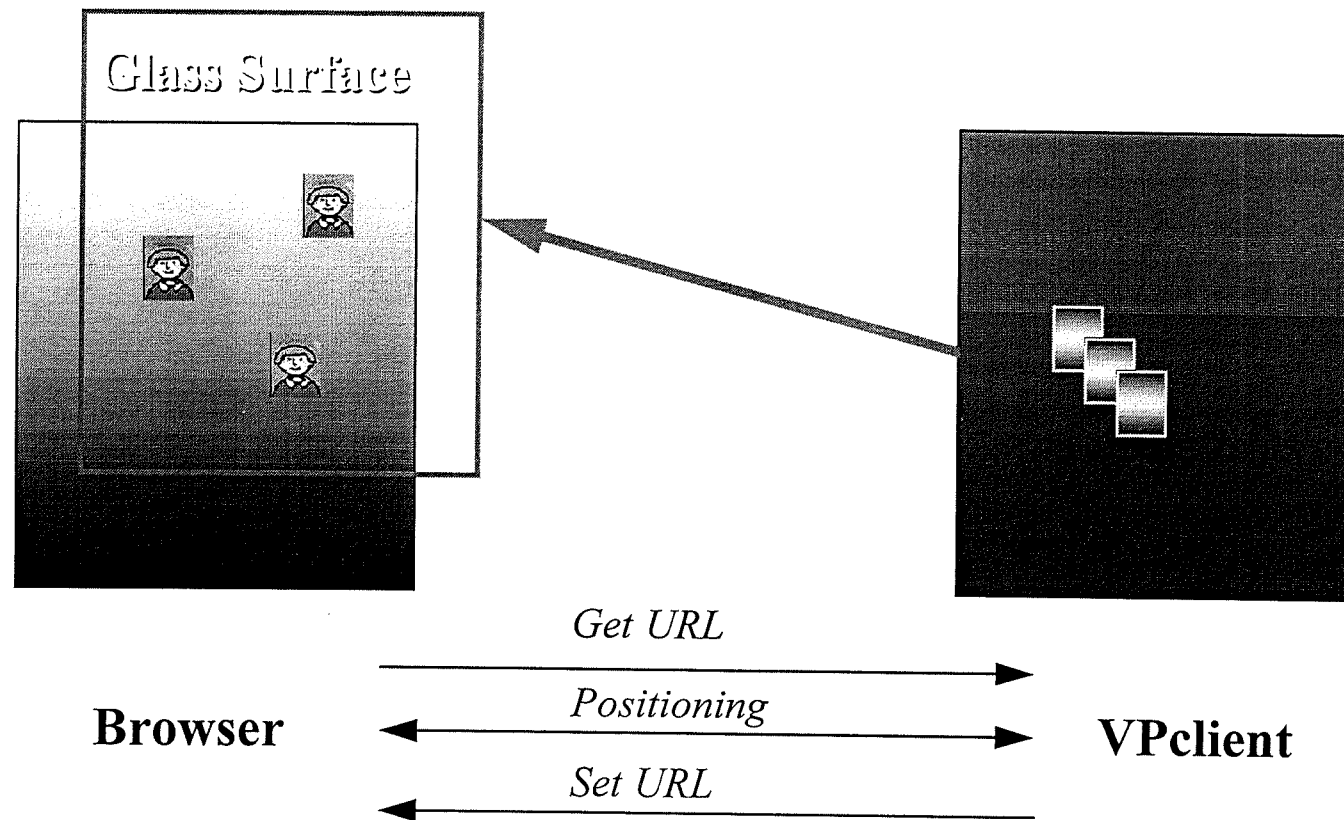
# *Client Architecture*



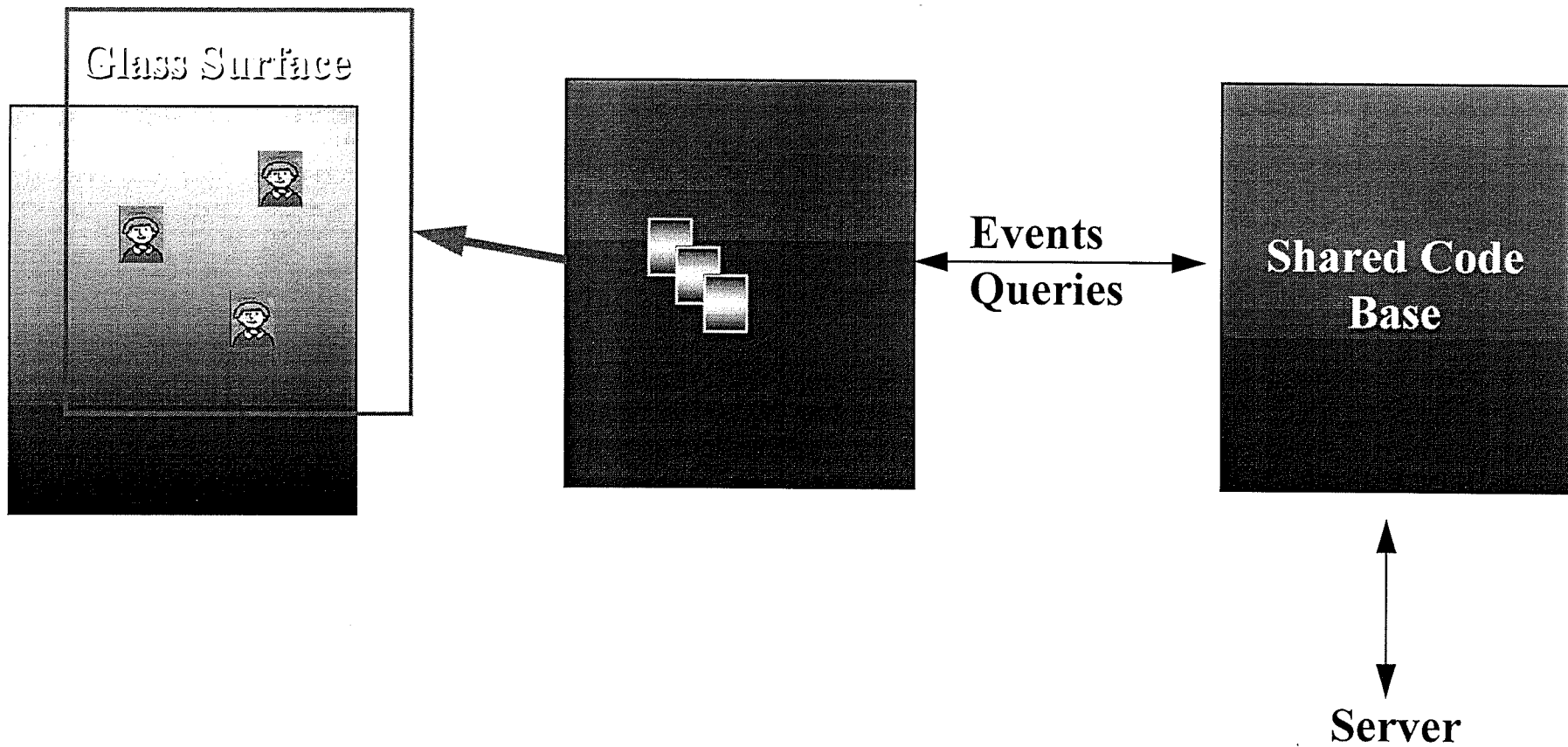
# *Client Architecture*



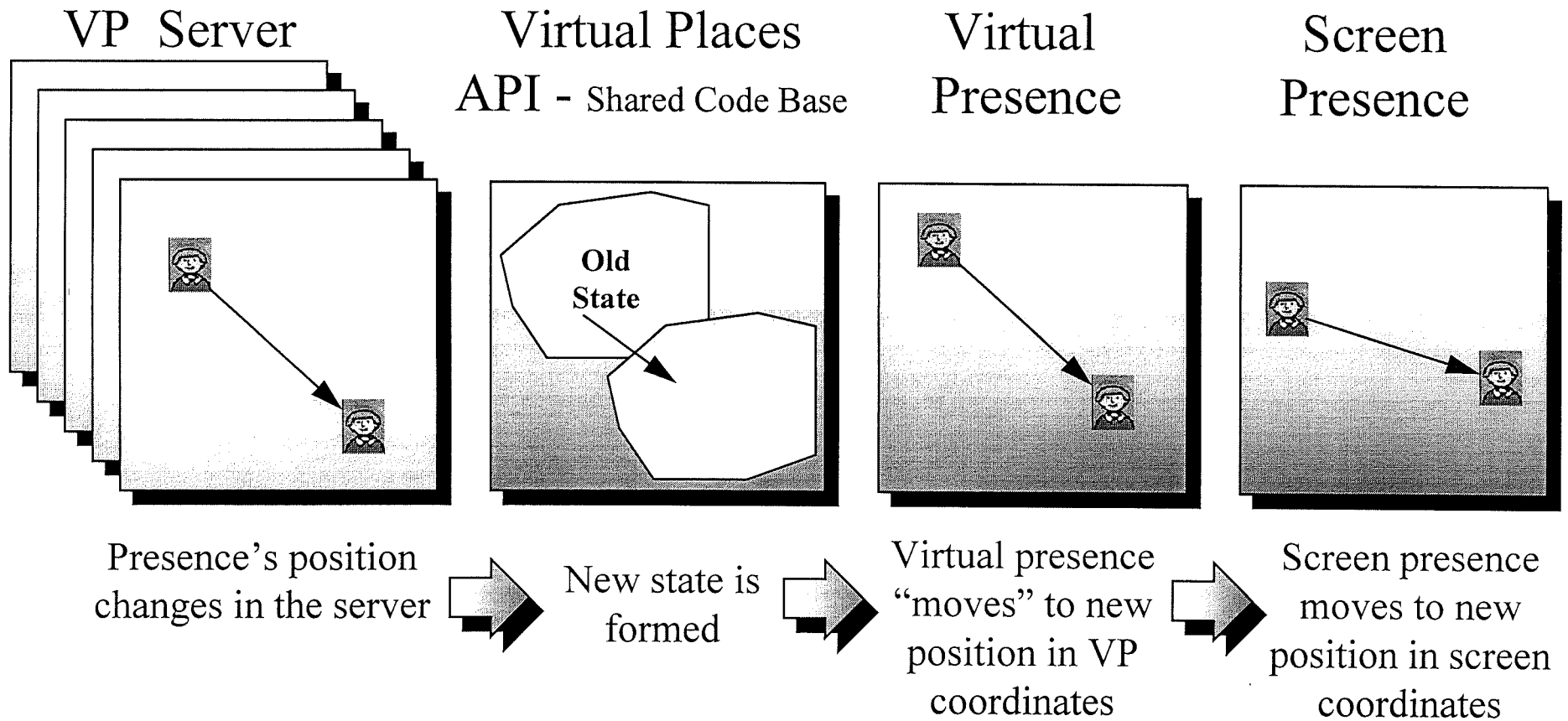
# *Client Architecture*



# *Client Architecture*



# Typical Event Flow



# *Client Architecture*

**Two implementations models to add to VP-enable a browser:**

- ❑ By hooking - Operating System Integration**
- ❑ By integration - Application Integration**

# *Two Implementation Models*

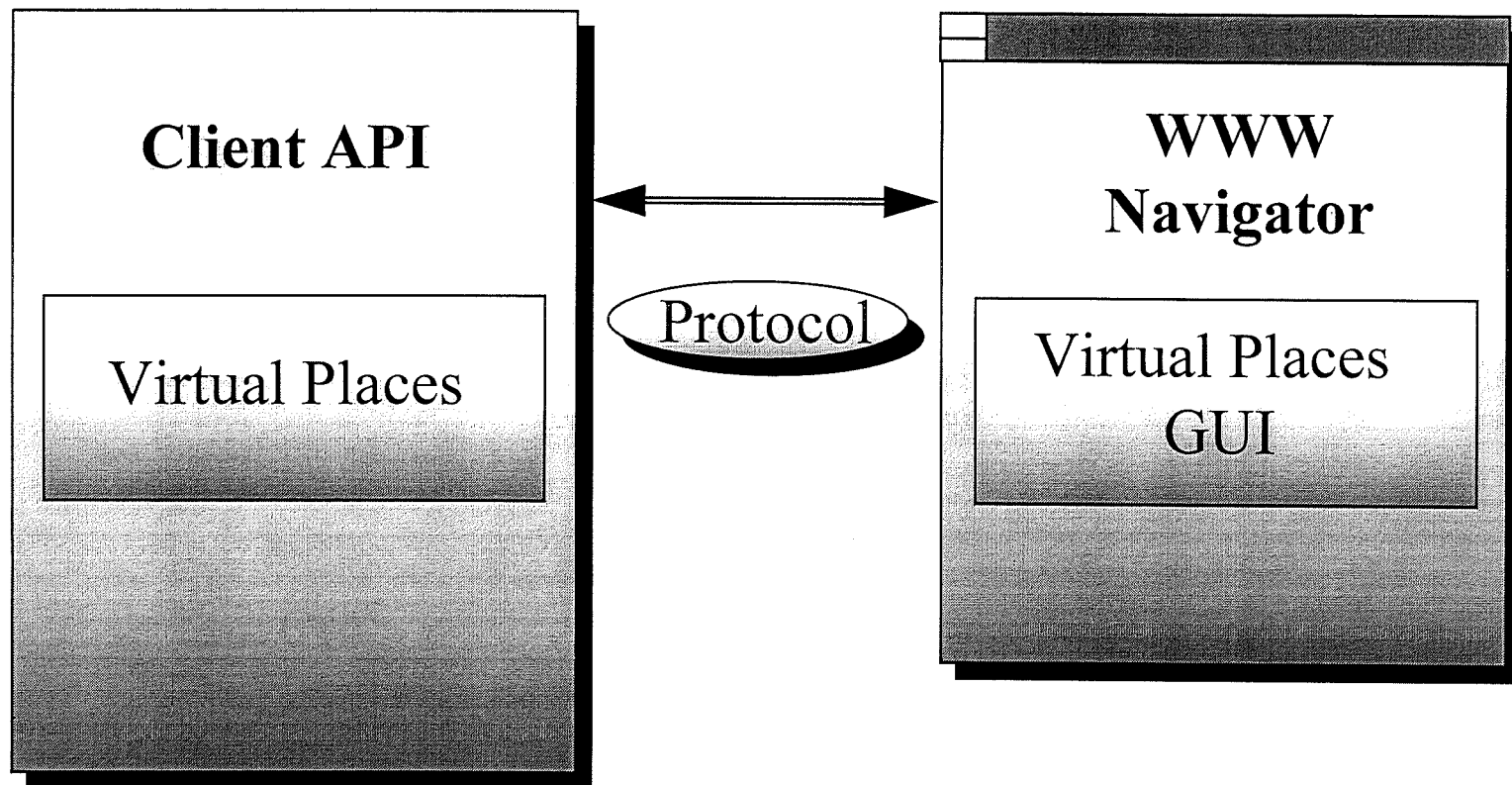
## **□ By Integration**

- WWW browser Links to a DLL which implements the screen presences
- The VP tool box is an integral part of the browser GUI

## **□ By hooking**

- All VP activity is implemented as a *separate application*.
- The communication between the VPclient and the browser (URL, Positioning) is done by using conventional (Navigator API, SDI) or non-conventional (Hooks) methods

## *By Integration*





# *Minimal VPCClient-Browser protocol*

## *URL*

**UINT Connect (URL) - Set VPC place (URL).**

**UINT GoTo(URL) - Set the Browser URL.**

## *Positioning*

**void AttachTo(window) - The window in which the VPC should render.**

**void SetSize(size) - The size of the attached window (document).**

**void SetOffset(size) - The offset of the attached window.**

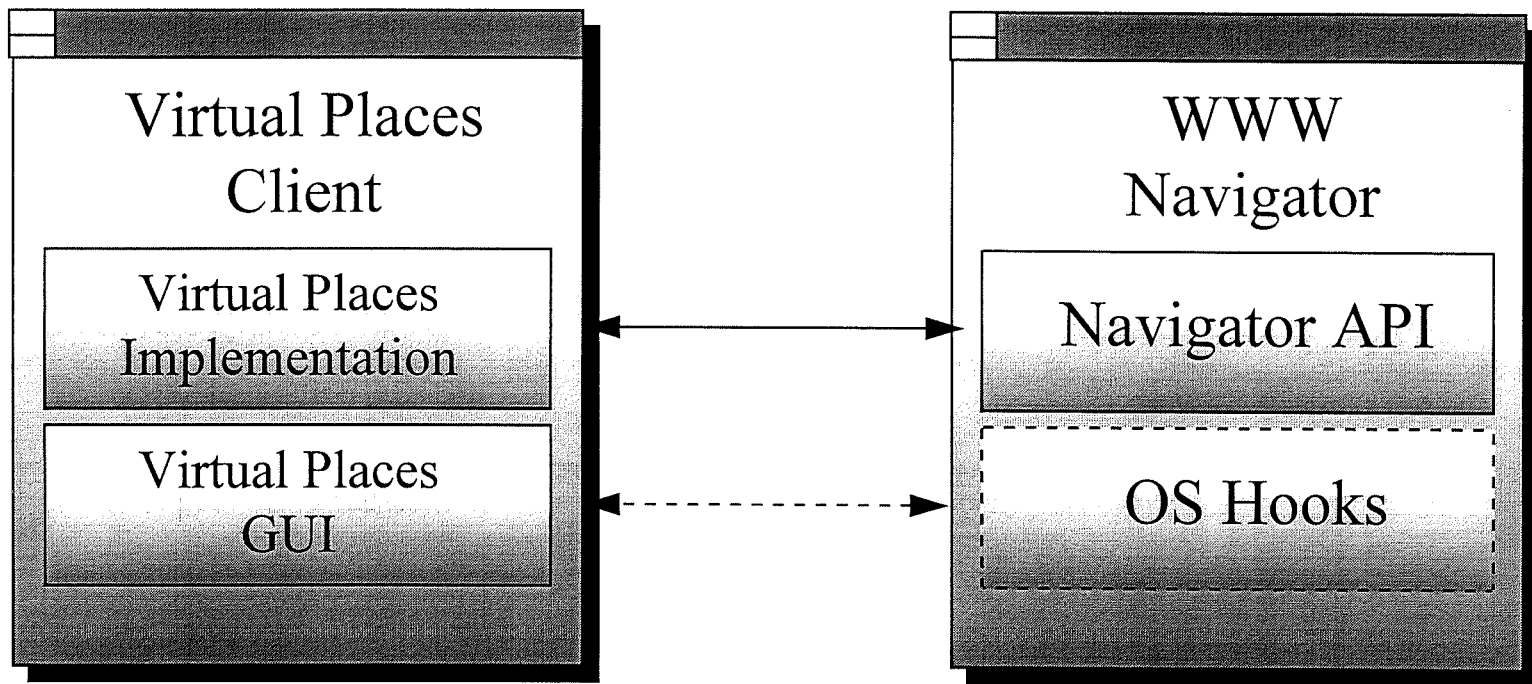
**void Refresh()**

## *Control*

**UINT Initialize() - Initialize the dll.**

**UINT Terminate() - Terminate the dll.**

## *By hooking*



*Navigator API: CCI (Mosaic), SDI (Spyglass, Netscape)*

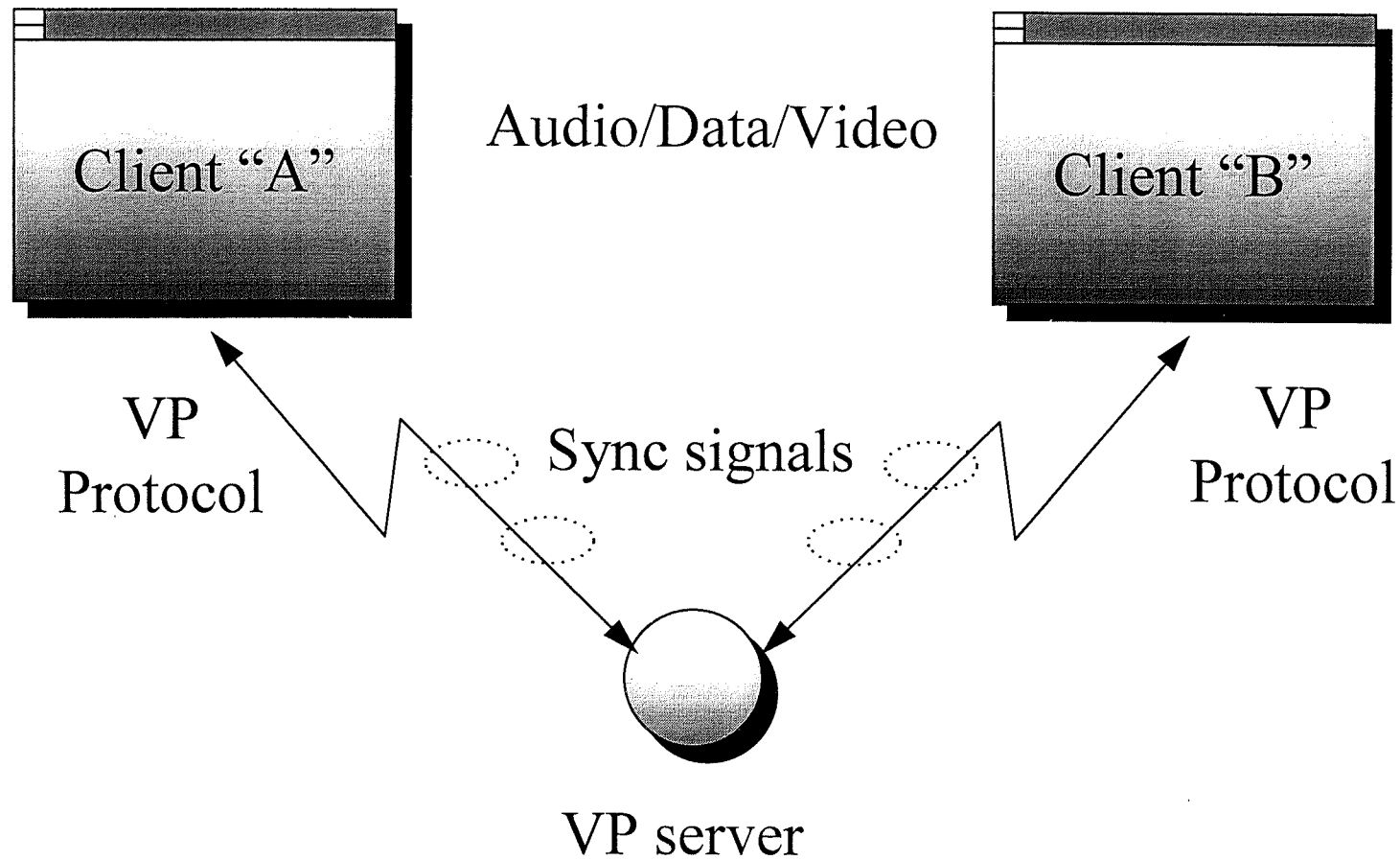
# *Methods Comparison Table*

<i>Entity</i>	<i>By Integration</i>	<i>By hooking</i>
Applications	1	2
Integration Effort	Major	Minor
Positioning	Document relative	Viewport relative
OS dependency	Minor	Major
Extensibility	Low	High
UI	Re-design	Standard VP design

## *Implementation Notes (PC)*

- ☐ Object-oriented methodologies
- ☐ 32 bit programming model (16 bit -possible) with respect to Win32s
- ☐ Implementation using a C++ Framework (MFC)
- ☐ Visual C++ 2.X
- ☐ Remote Windows 3.1X debugging

# *Client-to-Client Connections*



# *Ubique R&D - Structure & Tasks*

- ☐ **Virtual Places - Architecture**
- ☐ **Virtual Places - Client**
- ☐ **Virtual Places - Technologies**
  - **Audio**
  - **Video**
  - **Animation**
  - **VRML**
  - **Intelligent Agents / Extensions**

**Support**

**Technical  
Communication**

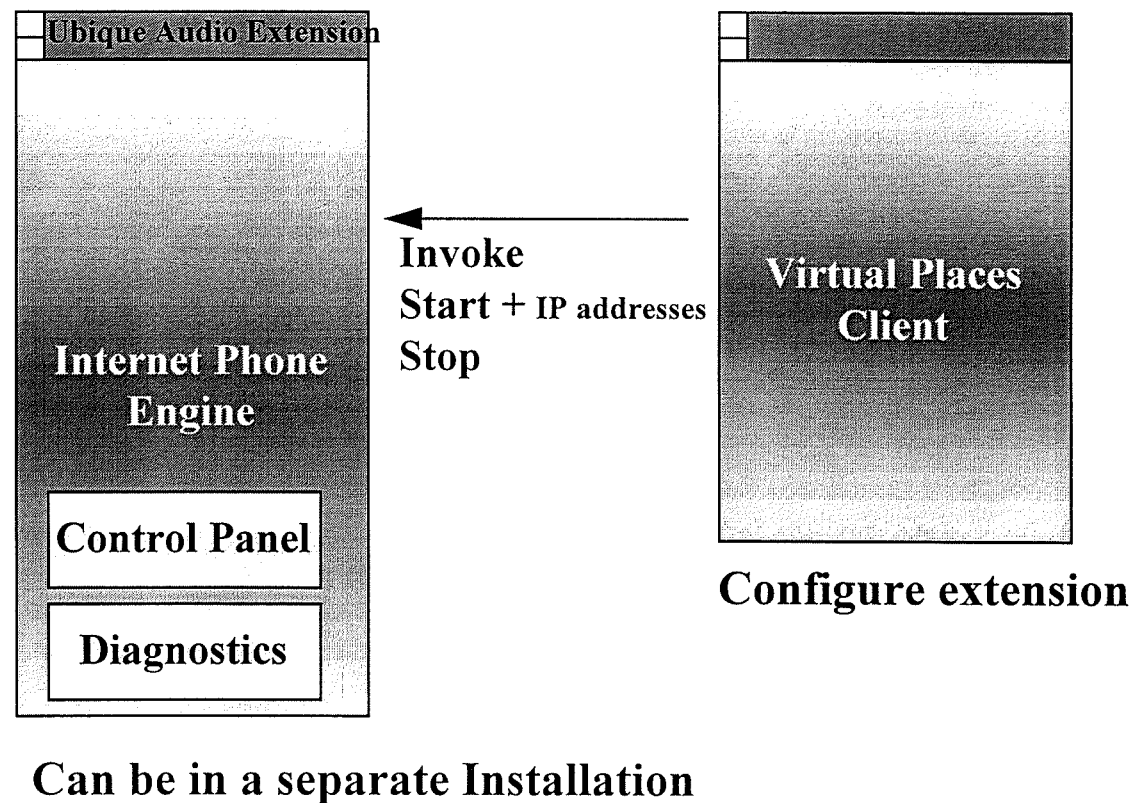
**QA**

# ***Technologies & Extensions***

**Virtual Places is a natural platform/application environment which enable to easily extend human capabilities in a virtual place**

- ❑ Audio/Video**
- ❑ Animation - human gestures**
- ❑ Extensions (Intelligent Agents)**
- ❑ VRML (3D Clients)**

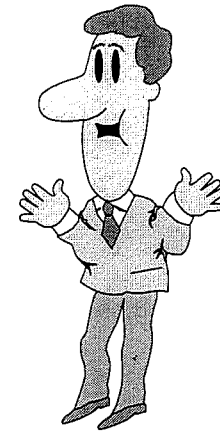
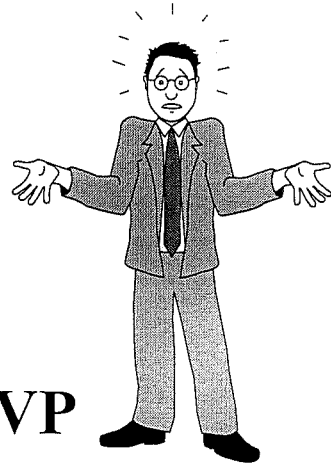
# *Audio (Video) Architecture*



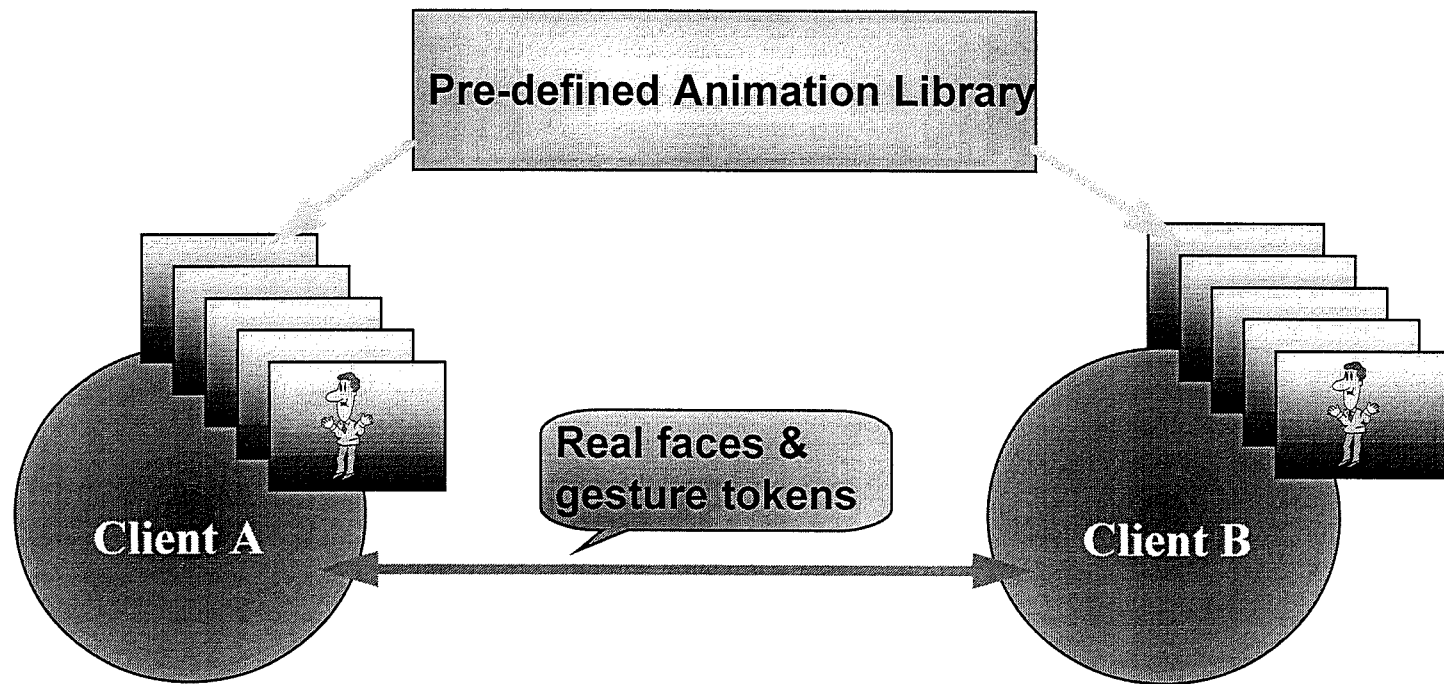


# *Animation*

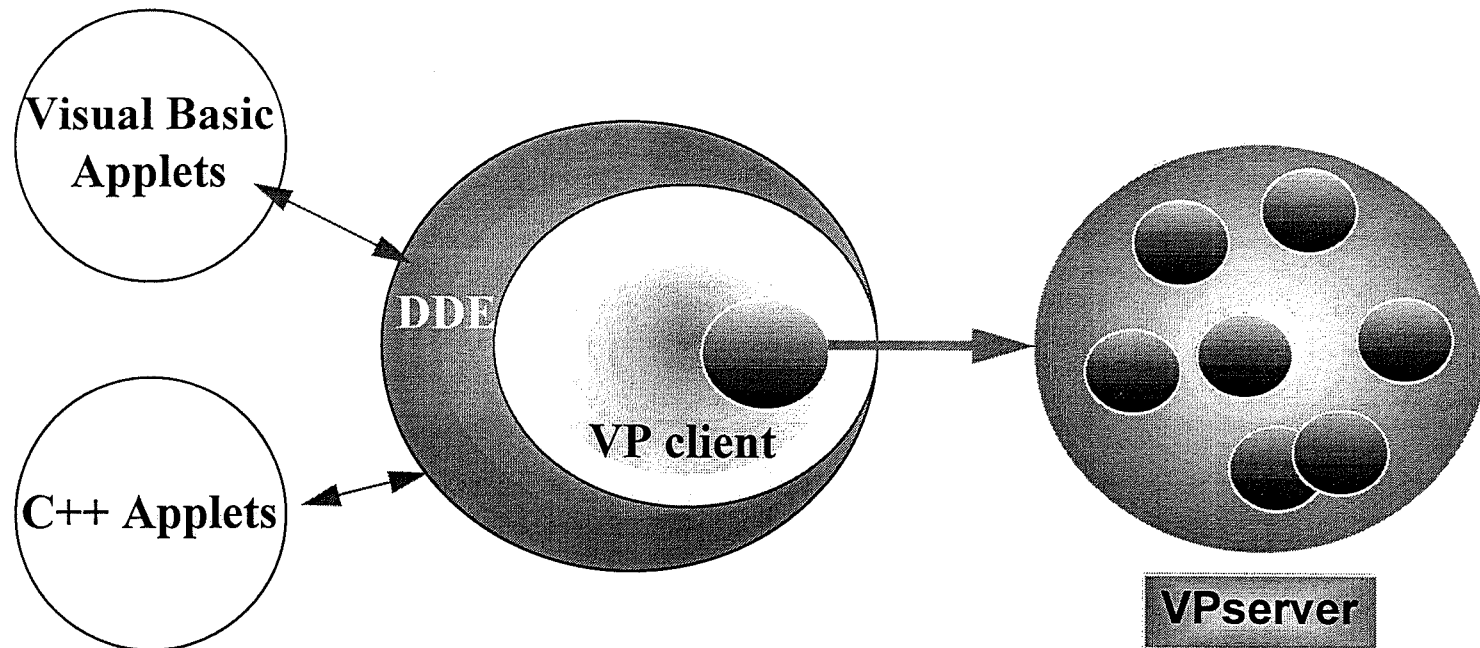
- ☐ Enrich gestures
- ☐ More fun / games
- ☐ Implemented above VP API
- ☐ Prototyping



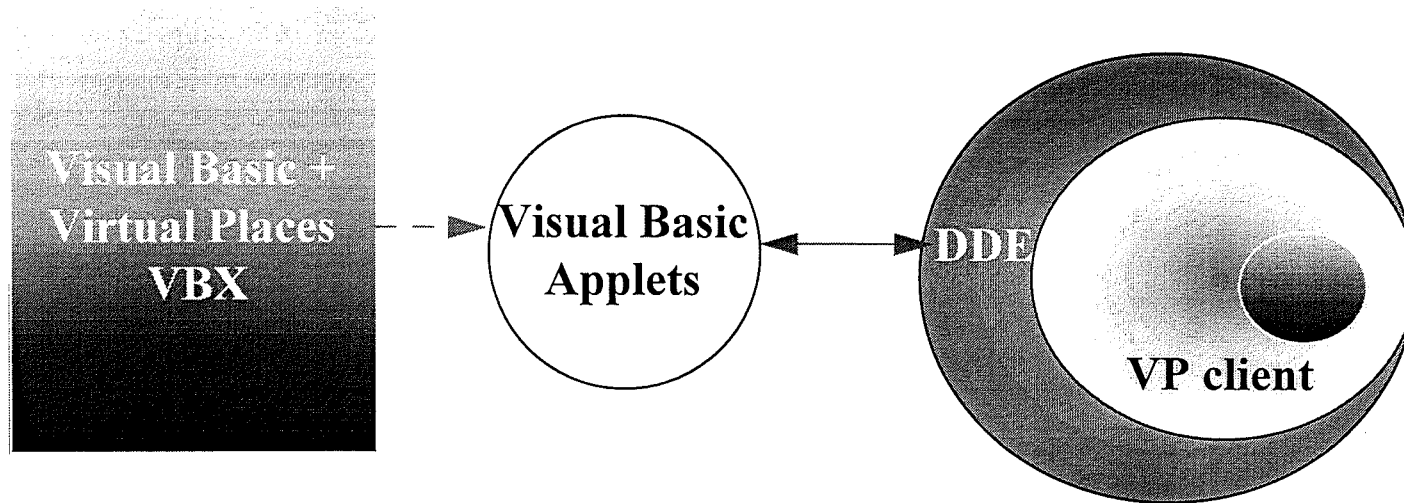
# *Animation*



# *Extensions - scheme*



# *Extensions - Visual Basic*



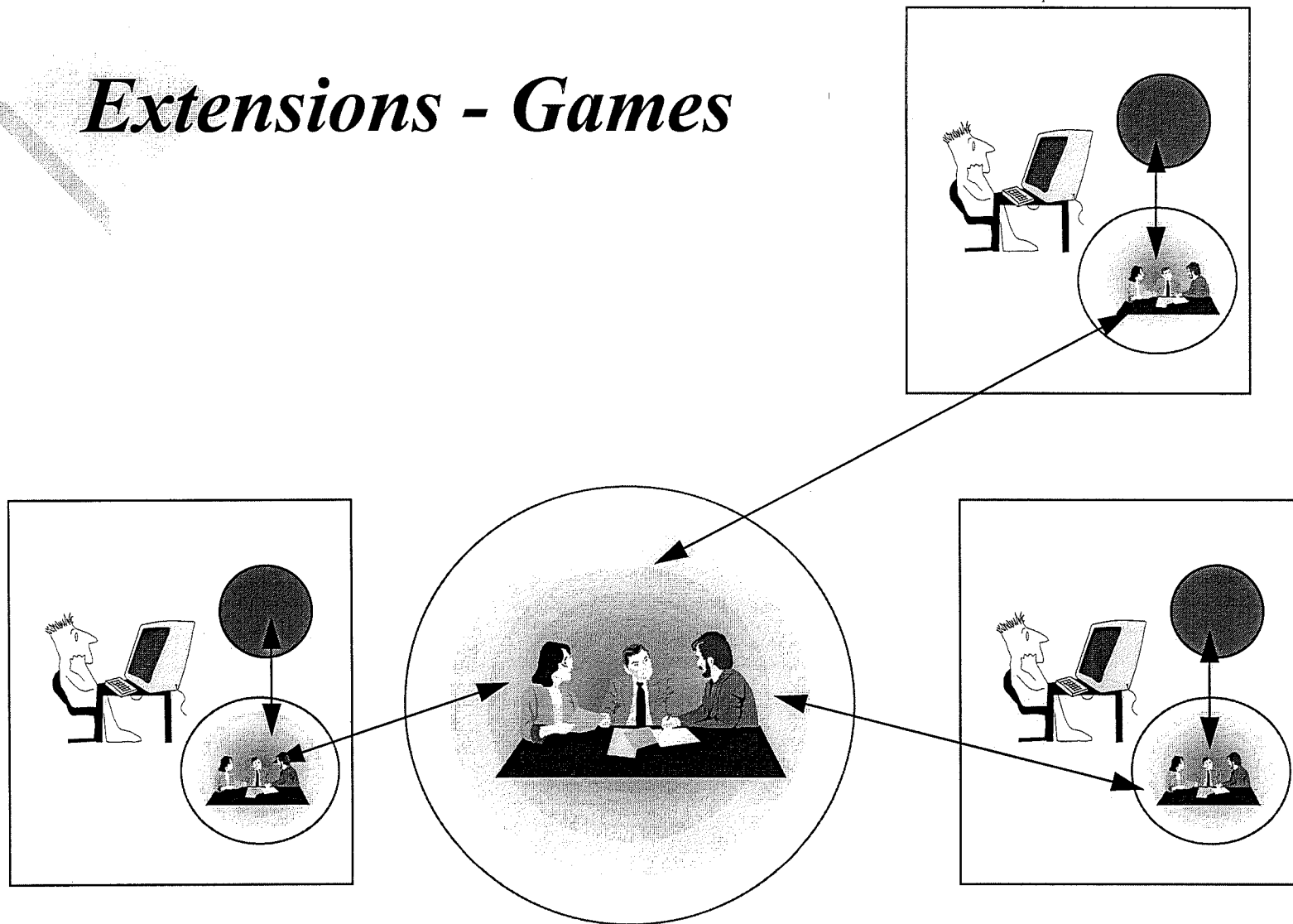
# *Visual Basic Extensions - Examples*



- ✓ Guided tours
- ✓ Graphical scriptor for intelligent agents
- ✓ Answering machines
- ✓ The Virtual-Press-Conference Kit
- ✓ Games: Bridge
- ☐ Limited natural language processing
- ☐ Screening
- ☐ Canned interactions

Status: Prototype level

# *Extensions - Games*



# ***VRML***

- ❑ Virtual Places for VRML requires tight integration with VRML Browser**
- ❑ VP/VRML API requires injection and manipulation of 3D objects in a VRML scene**
- ❑ Ubique is involved in creating an API for interactive VRML (2.0)**
- ❑ A prototype, based on WebSpace (SGI platform) and VP-API is under development.**